

# Projet Système d'Information

Rendu phase 2

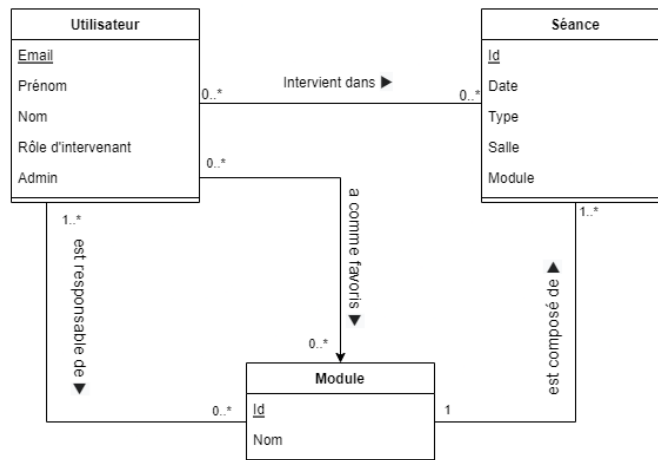
Groupe OEM

# MCD

Dans notre MCD nous trouvons 3 entités qui permettent, ensemble, de traduire le besoin client :

## Rôle

Prof / Doctorant / Exté



## Type

TD /CM / TP /TA

### - Entité “**Utilisateur**” :

Cette entité représente les utilisateurs de notre application, avec comme attributs l’adresse mail, le prénom et le nom qui sont des champs textuels. Le rôle d’intervenant est une énumération avec comme possibilité “prof”, “doctorant”, ou alors “intervenant extérieur”. Le dernier attribut “Administrateur” est un booléen qui détermine si l’utilisateur possède les droits d’administrateur ou non.

### - Entité “**Séance**” :

Cette entité représente une séance de cours. Une séance dure 1h30. Comme attributs, nous avons l’ID de la séance et le module qui sont entiers, la salle qui est un champ textuel, le type qui est de type énumération avec comme possibilités “TD”, “TP”, “CM” et “TA”.

### - Entité “**Module**” :

Cette entité représente un module. Un module est composé de plusieurs séances et a un ou plusieurs responsables. Il est caractérisé par un identifiant et un nom. Un module peut aussi être ajouté en favori par un utilisateur.

# Les Vues

## CAS - 1U

Paramètres : Authentification

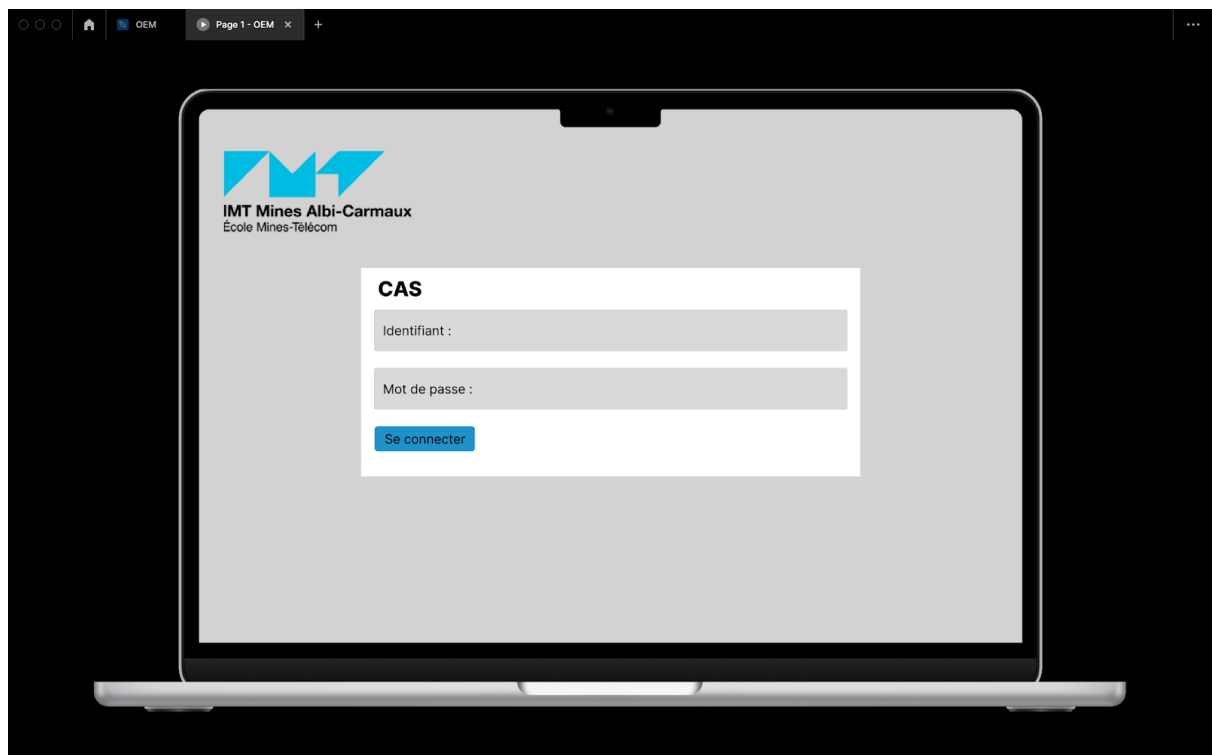
Accès : Tous

Composants :

- Zone d'authentification

Description :

*Cette page permet d'authentifier l'utilisateur grâce au CAS de l'école.*



## Accueil - 4U

Paramètres : Séances par groupe / Nombre d'heure / Type d'heure

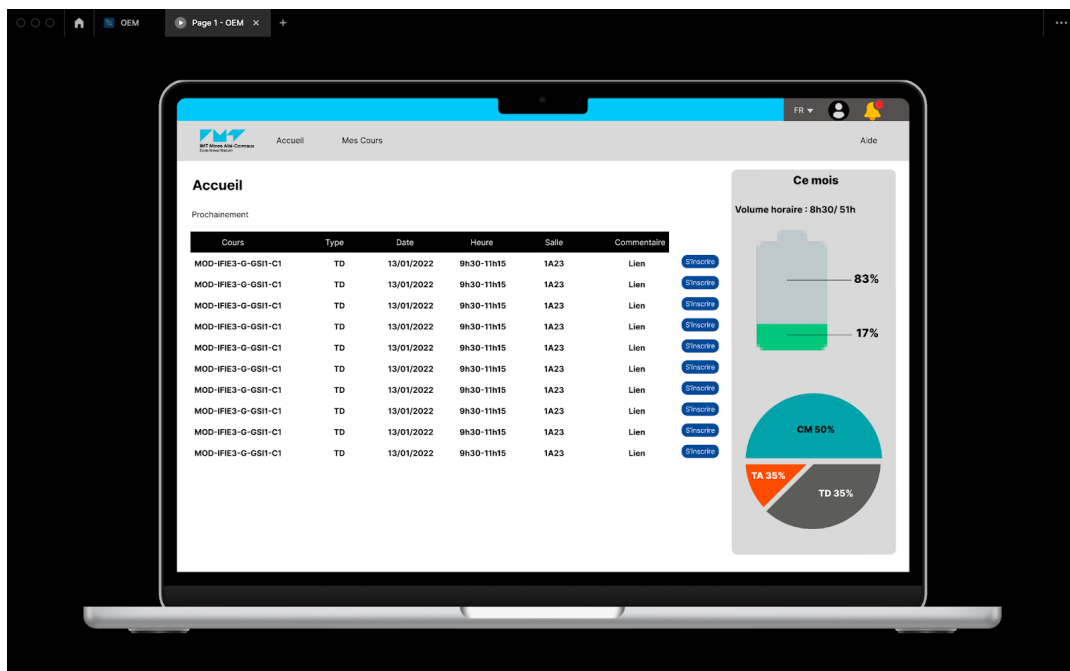
Accès : Tous les intervenants

Composants :

- Tableau des séances
- Dashboard sur le profil

Description :

*La vue accueil est découpée en deux parties. On retrouve dans la première les cours dont la date se rapproche (J- 7) et qui n'ont pas encore d'intervenants. L'utilisateur peut alors s'inscrire rapidement à un ou plusieurs cours proposés. La partie de droite comporte une représentation visuelle des heures de l'intervenant. On y retrouve le nombre d'heures inscrit sur le nombre d'heures restant.*



## Mes cours - 4U

Paramètres : Séance

Accès : tous les intervenants

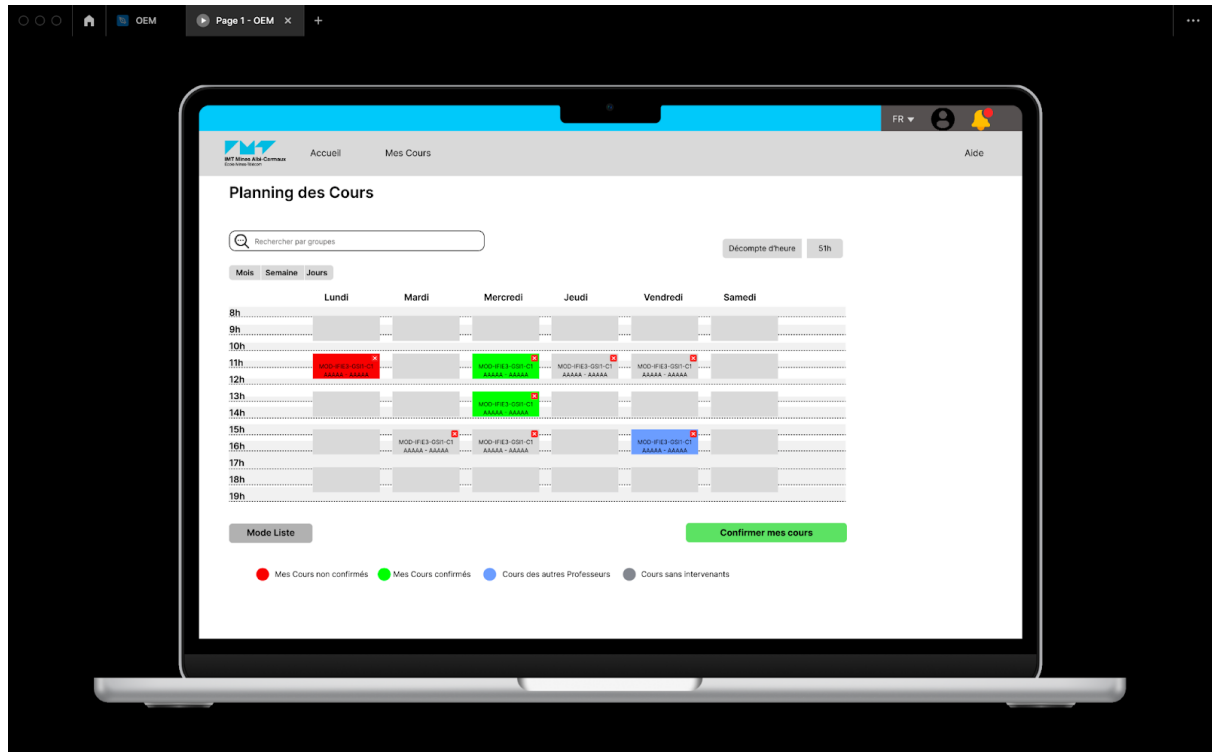
Composants :

- Information (ex : popup de confirmation d'inscription)
- Tableau avec la liste des séances, dates, type et intervenants inscrits

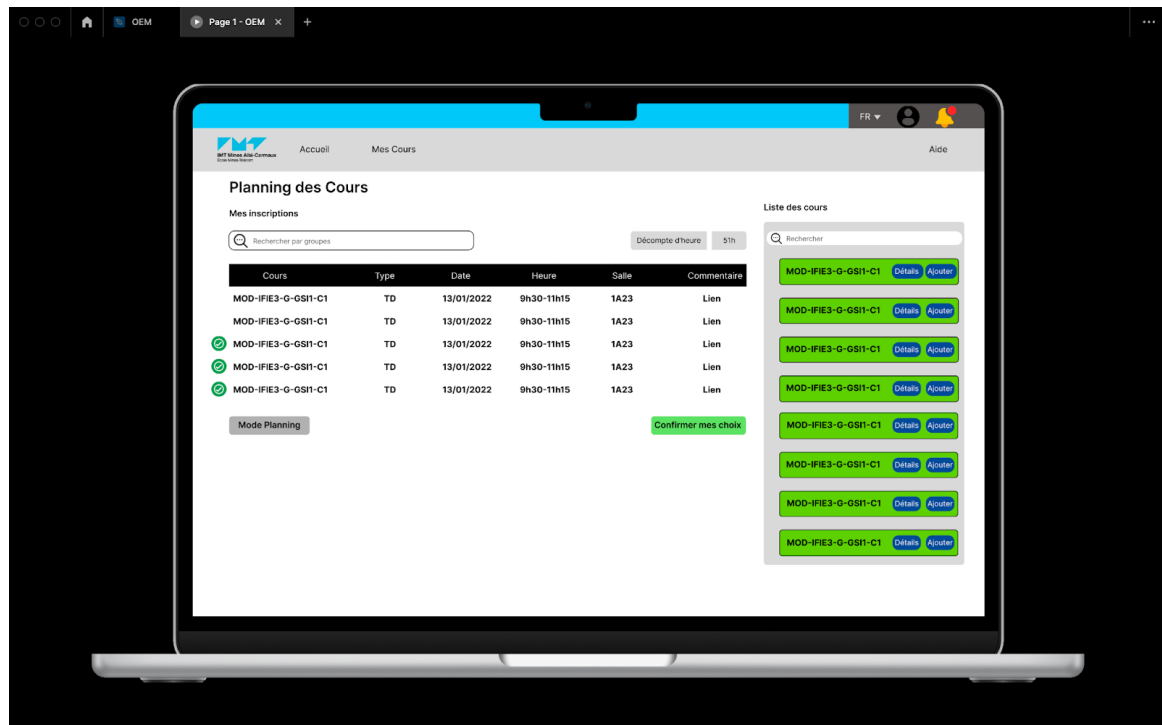
Description :

*Planning des cours (libres, occupés, que la personne doit donner) qui peut être fait en plusieurs affichages, sous forme de planning ou sous forme de liste.*

L'onglet "Mes cours" peut s'afficher sous deux formes. La première est en vue agenda, la barre de recherche permet de choisir un groupe d'élèves à poster. Le code couleur est décrit en bas du planning. L'intervenant peut directement s'inscrire en cliquant sur une case du planning. Pour enregistrer ses demandes il devra cliquer sur "confirmer mes choix"



La vue liste met en avant la liste des cours récemment inscrits pour ou par l'intervenant.  
 La section de droite permet de faire des recherches sur des séances et les rajouter par la suite.  
 Lorsque l'intervenant fera ses choix il pourra ensuite confirmer ses choix en les validant.



## Panel admin (gestion des rôle) - 2U

Paramètres :

- type (string): type d'utilisateur parmi "professeur", "doctorant" et "extérieur"
- rôle (string): "responsable\_module" ou "admin"

Accès : administrateurs

Description :

Cette vue permet de gérer les rôles des utilisateurs.

## Respo module - 2U

Paramètres :

- module (int): id du module

Accès : administrateurs

Description :

Permet d'afficher le remplissage des séances d'un module ainsi que la liste des intervenants et les heures qu'ils peuvent effectuer.

# Routes

Pour toutes les routes sauf login : l'utilisateur doit être authentifié.

## GET liste\_des\_séances

Permet de récupérer la liste des séances correspondantes aux filtres.

Accessible à tous les intervenants.

*Ne modifie pas la base de données.*

Paramètres :

- date\_min (date): date minimal
- date\_max (date): date maximal
- module (int): id du module
- type (int): type de séance
- libre (bool): séance libre ou non

Renvoie :

La vue liste des séances avec les filtres associés et aucun message

Charge : 4U

## POST ajouter\_intervenant

Permet d'ajouter l'intervenant d'une séance et donc de s'inscrire.

Accessible à tous les intervenants s'ils utilisent leur propre id ou aux responsables de module s'il s'agit d'une de leur séance.

*Modifie la base de données.*

Paramètres :

- séance (int): id de la séance à modifier
- intervenant (string): id de l'intervenant à inscrire.

Renvoie :

La vue liste des séances avec les filtres associés et un message indiquant la réussite de l'opération

Charge : 2U

## POST retirer\_intervenant

Permet de retirer l'intervenant d'une séance et donc de se désinscrire.

Accessible à tous les intervenants s'ils utilisent leur propre id ou aux responsables de module s'il s'agit d'une de leur séance.

*Modifie la base de données.*

Paramètres :

- séance (int): id de la séance à modifier
- intervenant (string): id de l'intervenant à désinscrire.

Renvoie :

La vue liste des séances avec les filtres associés et un message indiquant la réussite de l'opération

Charge : 2U

## GET role\_list

Permet de récupérer la liste des rôles des différents utilisateurs.

Accessible uniquement par les administrateurs.

*Ne modifie pas la base de données.*

Paramètres :

- type (string): type d'utilisateur parmi "professeur", "doctorant" et "extérieur"
- rôle (string): "responsable\_module" ou "admin"

Renvoie :

La vue de gestion des rôles, filtrée par les paramètres

Charge : 4U

## POST add\_role

Permet d'ajouter un rôle à un utilisateur.

Accessible aux administrateurs.

*Modifie la base de données.*

Paramètres :

- email (string): id de l'utilisateur
- admin (bool): administrateur ou non
- module (int): id du module

Au moins un argument entre admin et module doit être fourni



Renvoie :

La vue de gestion des rôles avec un message indiquant le succès ou non de l'opération

Charge : 2U

## POST delete\_role

Permet de retirer un rôle à un utilisateur

Accessible aux administrateurs.

*Modifie la base de données.*

Paramètres :

- email (string): id de l'utilisateur
- admin (bool): administrateur ou non
- module (int): id du module

Au moins un argument entre admin et module doit être fourni

Renvoie :

La vue de gestion des rôles avec un message indiquant le succès ou non de l'opération

Charge : 2U

## POST add\_favorite

Permet d'ajouter un module en favori à un utilisateur.

Accessible par tous les intervenants s'il s'agit de leur propre id.

*Modifie la base de données.*

Paramètres :

- email (string): id de l'utilisateur
- module (int): id du module

Renvoie :

La vue d'accueil

Charge : 1U

## POST delete\_favorite

Permet de retirer un module en favori à un utilisateur  
Accessible par tous les intervenants s'il s'agit de leur propre id.  
*Modifie la base de données.*

Paramètres :

- email (string): id de l'utilisateur
- module (int): id du module

Renvoie :

La vue d'accueil

Charge : 1U

## GET dashboard\_module

Permet d'afficher le remplissage des séances d'un module ainsi que la liste des intervenants et les heures qu'ils peuvent effectuer.  
Accessible aux responsables des modules.  
*Ne modifie pas la base de données*

Paramètres :

- module (int): id du module

Charge : 4U

## POST notifier

Permet de notifier un autre utilisateur.  
Accessible aux responsables de module afin d'inciter un intervenant à s'inscrire.  
*Ne modifie pas la base de données.*

Paramètres :

- email (string): id de l'utilisateur
- message (string): message à envoyer

Renvoie :

La vue de gestion de module.

Charge : 2U

## POST login

Permet d'authentifier un utilisateur.  
*Ne modifie pas la base de données.*

Paramètres :

- email (string): id de l'utilisateur
- password (string): mot de passe

Renvoie :

La vue d'accueil

Charge : 4U

# Choix de développement

## Capacité de production

Nous disposons de 108 h de développement chacun. En ajoutant 10% de temps personnel, nous arrivons à 118,8 h par personne, soit une capacité totale d'environ 475 h.

## Charges

Charge totale des vues : 13U

Charge totale des routes : 28U

On considère qu'il faut rajouter 1U de test par vue et par route pour les tests. On obtient ainsi une charge de 16U.

On considère qu'une unité correspond à 10h de développement, soit une charge totale de 570h.

On estime le temps de réunion hebdomadaire à 3h, soit 72h sur les 6 semaines de développement pour notre groupe de 4.

La charge totale est donc estimée à 642h.

## Choix de développement

La charge étant supérieure à notre capacité de développement, nous allons prioriser certaines parties :

- Haute priorité : routes et vues liés à la connexion et à l'inscription des intervenants
- Priorité moyenne : routes et vues liés à la gestion des modules et à l'administration
- Priorité basse : route et vue liés à l'accueil et à la fonction favori